Considering Operational Security Risk during System Development

The operational security of software-intensive systems is closely linked to the practices and techniques used during system design and development. The authors examine OCTAVE, an operational security-risk methodology, and apply it to the security-related risks identifiable while developing software-intensive systems.

CAROL WOODY AND CHRISTOPHER ALBERTS
Software Engineering Institute

Software products today are riddled with defects, some of which leave systems vulnerable to cyberattacks. Although high-quality development processes can limit vulnerabilities, these processes alone aren't sufficient for operational security. Development processes must therefore explicitly address the security-related risks inherent in operational environments. Not only does this ensure that these systems maintain good operational performance, but it also reduces the risks to the business processes they support.

An effective process for identifying and addressing operational security risks is a security-risk assessment.[1] By applying selected steps from such an assessment during development, system developers can characterize the target system's potential security risks and gain a better understanding of the organization's risk potential. These steps must be inserted into the system-development methodology at multiple points, though, because operational considerations are affected, at a minimum, by decisions made during requirements elicitation, vendor selection, architecture design, component acquisition, and integration. Developers should also consider the security implications of choices made at different strategic development life-cycle points. If the anticipated operational security risk level is too high, developers can then consider additional options to mitigate the potential risks earlier in the development life cycle, rather than ignoring the risk and limiting responses to options available after the system is deployed.[2]

At the Carnegie Mellon Software Engineering Institute (SEI), we developed a security risk methodology called OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation) for evaluating and planning ways to address operational security risks.[3] It characterizes operational threats as security events triggered by people inside and outside of an organization, system problems, and problems outside of an organization's control, such as floods and power outages. The people component here includes authorized users such as technology staff, system users, managers, contractors, customers, vendors, or partners, as well as anyone exploiting unauthorized access from inside or outside the organization. Security components can include an organization's policies for acceptable technology use as well as firewall, antivirus protection, and access-logging software. In this article, we discuss OCTAVE within the context of analyzing an organization's potential operational security risks for a software-intensive system development project prior to actual deployment.

Specific attention

System development focuses primarily on functionality to meet user requirements—security isn't always a priority. A system's architectural design determines its qualities (of which security is only one).[4] Research confirms that quality attributes critical to the architectural design effort are difficult to define and hard to identify in system-validation steps.[5] Operational security is defined as the absence of security failure, but security failures manifest themselves once the system is placed in an operational environment and subjected to potential abuse from internal and external sources.[6] Based on this definition, clearly describing security requirements that unambiguously represent the absence of a failure in a way that can be confirmed through some form of verification is extremely difficult, and system developers can easily overlook their responsibilities for operational security.

Moreover, requirements, design, and coding errors can create security gaps that operational security can't completely address. These types of errors have always been in software and systems, but techniques to exploit them are fairly recent phenomena. As implemented systems' complexity increases, so too do the tools used to exploit them. Analysis reports from the CERT Coordination Center indicate substantial growth in the level of impact for each major attack due to the increasingly sophisticated attack tools available to a wide-ranging participant base.[3]

Implemented systems are collections of components linked together to share data and perform specific functions. These components can include vendor-supplied tools such as database-management systems, document-management systems, business-rules engines, and reporting tools as well as custom-developed software. Vendor-developed products such as accounting and billing packages might be combined with Web pages, remote mounted files, and interface modules to support business processes. These components might also be distributed across several operational environments and provide functionality to local, remote, and mobile users. Each component has a set of features and errors that let security failures occur if triggered by the right set of circumstances and events. System designers and developers must build the entire system to maintain appropriate security requirements across all of its components so that failure at any one point can't compromise all components.

The system must also work within the operational support environment in which it resides. Otherwise, techniques the operational support staff use to perform their roles might counteract internal system mechanisms, resulting in an overall increased failure risk. The people handling operational support for this environment are responsible for maintaining an organization's technological infrastructure and communications. This role might be handled internally or outsourced to a managed service provider. Operational-security needs form a major part of the operations support role, but a vast inconsistency exists in the application of operational standards and practices.[7]

Operational support can't keep up with the increasing volume of vulnerabilities, leaving systems exposed unless they're designed to meet specific security requirements and protected based on expected operational behaviors. Considering security risk earlier in the development life cycle provides system developers with the opportunity for lower cost-mitigation options, more stable solution options that can be carried into future development upgrades, and the opportunity to establish a consistent approach to security among developers, users, and system maintainers.[2]

A key aspect of improved operational security is establishing what security means to an organization. Terms such as confidentiality, availability, and integrity hold different meanings for organizational management, system users, and operational support staff, resulting in differing opinions as to what really is a security risk. From a management perspective, for example, system availability usually means authentication and authorization. To users, ease of use, accessibility, and response time are the primary characteristics of system availability. To support staff, system availability means network component and service uptime along with site-disaster recovery. System developers can miss key aspects of availability if they don't consider all perspectives in their decisions about availability requirements.[6]

Development and risk

Describing the target operational environment can be complex. New systems are often aimed at changing an organization's working environment, and the impact of these changes must be characterized and analyzed for effective security planning. OCTAVE uses a structured sequence of activities conducted by an interdisciplinary assessment team responsible for collecting the appropriate content and analyzing the security implications based on available information. Operational security experts, system architects, organizational management, and users must be represented in the team composition.

When assessing an operational system using OCTAVE, the assessment team examines the following area of potential operational security risk: security awareness and training, security policies and regulations, collaborative security management, contingency planning and disaster recovery, physical security, system and network management, system administration tools, monitoring and auditing, authentication and authorization, vulnerability management, encryption, security architecture and design, and operational staff security.[8] Although the development process addresses only a subset of these implementation security risks, system security actions must be designed to augment and not compete with the implemented operational security practices. Network security, for example, might not be understood prior to deployment, but the assessment team should consider the need for data encryption requirements if sensitive information is distributed to mobile devices—even though this decision might slow down system response.

System designers and developers must build the entire system to maintain appropriate security requirements across all of its components.

In addition to potential implementation security risks, a business process's migration to a new technology environment might introduce increased security risks. By transitioning a paper-intensive process to one that's technology controlled, for example, or shifting a business process from one platform environment to another, the opportunity for security threats to data confidentiality, integrity, and availability changes substantially. Such changes must be reflected in the software functionality, system implementation choices, and operational practices the assessment team proposes to address security risks.

New systems are often aimed at changing an organization's working environment, and the impact of these changes must be characterized and

Additional security risks are introduced via decisions made in the acquisition process that the target system inherits. Unfortunately, incorporating commercial off-the-shelf (COTS) products adds not only security features but also vulnerabilities and vendor limitations to the security-planning mix. The selection of an operating platform, database, and user interface inserts additional security features and limitations into the plan. Naturally, the use of third-party and possibly off-shore resources to support development and operational support also impacts the operational security risk.

Security risk in the development process

Using OCTAVE, the team can perform the following steps:

- define the target system to be implemented and identify the information assets the system creates, changes, and uses;
- determine each information asset's security attributes (confidentiality, integrity, availability) and how the target operational system accommodates them for each information asset in the system;
- identify potential security threats to the target system and how they affect information assets;
- identify the security risks linked to outcomes based on whether the threats identified in the previous step actually materialize; and
- develop a protection (risk-mitigation) plan comprised of component and procedural protections to address and mitigate unacceptable security risks.

The depth of security analysis and planning an assessment team can achieve will vary based on the amount of detail in the target system's description. As a result, this sequence of activities should be applied at critical milestones throughout the development life cycle to ensure appropriate consideration of operational security risk. Our experience has shown that an assessment should be conducted in preparation for the following critical decision points: requirements acceptance, architecture acceptance, and integrated system validation.

Define the target system

To define the target system, the assessment team describes each system component in as much detail as is known. As the development process moves closer to completion, a clearer picture of the target system emerges. The assessment team should also characterize the information the system handles and the users who interact with the data. Our experience has shown that this characterization has greater detail than initial system security requirements and should be used to confirm and improve the development of security requirements.

It's imperative that the target system's definition include all planned software and hardware components. These might include standard operational platforms, current client-server configurations, shared firewalls, and intrusion detection mechanisms, as well as system-specific COTS modules under development. If the assessment team can evaluate security risks prior to purchasing the final components, those handling the acquisition contracts can apply the risk considerations to the selection process—ultimately, the target system inherits each purchased component's security risks.

Analyze security attributes

To determine the target system's security attributes, the assessment team must analyze the critical information stored, processed, and transmitted in and by the system. This information should be a reflection of the system's security requirements. Unfortunately, most security requirements we've seen are too general—it's not enough to merely require the system to prevent unauthorized access—specifying who's authorized and who isn't must be unambiguous and verifiable.

The security attributes for patient data in a medical system, for example, include confidentiality, integrity, and availability. Only authorized people can access the data, but authorized people can include a wide range of roles such as medical providers, insurers, billing staff, and the actual patient. Such data must have a high integrity to assure appropriate care for the patient, thus only authorized people can modify this information. In addition, access to patient data must be available at all times for any possible patient encounter. These descriptions of security attributes aren't sufficient without clarifying how authorized individuals will be identified and authenticated, how authorization and patient information will be shared across the target system's components, and how authorization and patient data will be handled in each component. Each step in the dataflow across system components and interfaces must consistently support security requirements in the same manner. Gaps and discrepancies can become potential vulnerabilities.

Identify threats

In identifying threats, the assessment team must consider who or what could compromise a target system's components such that the system's security attributes would be jeopardized. Although this might appear similar to a general operational threat analysis, identification in this step should focus on ways in which the target operational system, information assets, and components within it differ from what already exists in the current operational environment.

The operational environment in which a system is implemented is a key contributor to external threats, many of which might be sufficiently addressed by existing operational practices. However, if the system's information assets differ greatly from those the operational environment currently protects, existing practices might be insufficient; this gap must be identified as a threat. If the target system requires wireless external access that differs from the existing operational system, for example, then current operational security practices might not be sufficient and the system could be vulnerable to threats.

The assessment team must also consider the available operational capabilities for identification and recovery from a compromise and how these fit into the existing operational environment. If an application is designed assuming a two-hour recovery period from a system outage, but the operational environment is set up for an eight-hour response to site failure, this disconnect represents a threat to the system.

The system's construction might also increase its vulnerability. Some questions the team might consider include the architectural choices made about the system's ease of use, performance, and security; whether the system structure requires centralized or distributed operational management; and what decisions, if any, have been made for data replication.

Interfaces represent another area ripe for compromise and interface design and implementation decisions might represent potential threats. The assessment team must consider the critical interface relationships the target system has, the information that's being shared, and the ways it's protected in transit, as well as the established trust relations between systems, processes, and components and the ways in which trust is communicated.

Another area of potential threat is the implementation process. Are there possible implementation errors, or are there limitations of selected implementation iterations that would require data to be moved back to the old environment during the transition to production on a case-by-case basis? Can this return be handled automatically, or is there an assumption that system functional owners will use a manual procedure and is this feasible? A recovery failure represents a threat.

Identify risks

Identifying security risks means considering each threat's potential impact on the organization. If the target system is compromised and critical information is disclosed, modified, destroyed, or inaccessible for a length of time, how would the organization handle it? This impact can be magnified if the new system's operation becomes unstable. Impacts can include

- inability of the system to complete business functions;
- revenue loss;
- increased operational costs due to overtime payments or other additional expenses;
- reputation damage;
- fines and penalties; and
- safety and health problems for system users.[2]

The expected impact of the new system's implementation on the organization's existing business operations plays a factor in this analysis: will users juggle dual-system work as they learn new procedures and business practices? How will they notice and react to operational problems with the system? Will such problems put business processes at risk?

An assessment team can evaluate the likelihood of external threats if it knows the target organization's technology infrastructure histories with cyberthreats, similar organizations' histories with cyberthreats, or similar systems' histories with cyberthreats. Histories should include information about how often such attacks have occurred in the past as well as the severity of the resulting impacts on the organizations. In many organizations, operational support people collect this information but most haven't thought about making it available to system developers.

Each step in the dataflow across system components and interfaces must consistently support security requirements in the same manner.

The assessment team should also consider the history of an organization's project development success. Is there a history of vendor-implementation problems? What have other organizations experienced with COTS products and components selected for this target operational system?

Create a plan

Addressing operational security risks requires planning. Because security is a shared responsibility, the assessment team's participants in this effort must include those people who can identify protection opportunities—operational staff, system users, developers, those responsible for components or interfacing systems, and senior management. The team can segregate protection into two parts based on the way in which it must be implemented: component protection and procedural protection.

Components to consider in the plan include software applications, hardware, operating environments (such as the operating system, I/O management, and networks), COTS components, and system interfaces. Operational support resources for each of these might differ, so the team must blend mitigation efforts with existing operational efforts. The team should also review the full suite of operational security risks and consider which ones are important to each component and how to address each component's security. Some key questions to debate might include whether to use encryption to protect information while in storage and during transmission and how to maintain security levels if the organization upgrades components.

Procedural protection areas include the steps required for transition from development to operational environment, operational disaster recovery and continuity plans, and access-control management. The team should review and consider several procedural areas:

- how hardware and software components are locked down (hardened) during the transition from development to implementation;
- the procedures in place for maintaining the target system beyond implementation;
- whether the new system's needs fit the current operational backup procedures and contingency plans;
- who monitors and manages access control for the target system and how they make decisions to grant or revoke access;
- how to control the access of contractors and vendors with support contracts and whether remote access will be used for this support;
- the training and support users will need to access and use the target system; and
- the procedures and support resources available to handle ongoing user support beyond the initial implementation.

At the end of the planning, some uncertainty will remain because not all risk can be eliminated, but this residual risk should be greatly reduced using this OCTAVE-based risk-assessment approach.

OCTAVE's focus on the operational environment differentiates it from other types of security assessments. Some security assessments focus primarily on code analysis and are used to eliminate highly visible vulnerabilities, such as buffer overflows. Other types of risk-based approaches, such as threat modeling, focus on establishing and testing requirements and ensure that specific threats are prevented. A third category of security assessments focuses on compliance. The US National Institute of Standards and Technology (NIST), for example, published a methodology that examines the extent to which a system complies with relevant statutes and regulations. None of these alternative approaches, however, examines an implemented system in the context of its current operational environment—a key aspect of the OCTAVE approach.

SEI successfully incorporated the OCTAVE methodology in several independent technical assessments (ITAs) of development projects for US government agencies such as the Internal Revenue Service (IRS), Veterans Administration, and Department of Defense (www.sei.cmu.edu/programs/acquisition-support/about.html). In each assessment, key stakeholders from management, operations, and development, including system architects and business subject-matter-experts, formed an assessment team to evaluate operational security and identify gaps in their organizations. OCTAVE's approach provided a mechanism for organizing the conversations that led, in two instances, to the discovery of major security design flaws. Because of early detection, the system developers had time to adjust their operational approaches without jeopardizing project timetables.

Additionally, the programming services group at Liberty University in Virginia applied this operational security approach to identify ways the development staff should tailor their system development life cycle (SDLC) to effectively address operational security risk.[9] From this work, we learned ways to

- gauge the importance of security to the development project owners;
- identify security risks based on past experience in the current operational environment;
- be specific with users about their roles and responsibilities in protecting the system and what the system will and won't protect; and
- build operational security knowledge within the development team because operational security experts have limited availability and any reliance on them can impact project timetables.

SEI released OCTAVE for public use to address operational security risk management in September 2001 (www.cert.org/octave/methods.html). By assigning an assessment team to apply the activities we describe in this article to systems and software still in development, organizations will have the ability to plan for and reduce operational security risk prior to operational deployment. ☐

References

1. C. Alberts and A. Dorofee, *Managing Information Security Risks: The OCTAVE Approach*, Addison-Wesley, 2003.
2. C. Alberts, A. Dorofee, and C. Woody, "Considering Operational Security Risks during Systems Development," *Software Engineering Process Group Conf.* (SEPG 2004), 2004; www.cert.org/archive/pdf/alberts-2.pdf/.
3. C. Alberts, A. Dorofee, and C. Woody, "OCTAVE Overview: Operationally Critical Threat, Asset, and Vulnerability Evaluation," white paper, Carnegie Mellon Univ./Software Eng. Inst., 2003; www.cert.org/octave/pubs.html.
4. P. Clements, R. Kazman, and M. Klein, *Evaluating Software Architectures: Methods and Case Studies*, Addison-Wesley, 2002.
5. N. Mead, D. Firesmith, and C. Woody, "Eliciting and Analyzing Quality Requirements: A Feasibility Study," tech. report, CMU/SEI-2004-TR-018, Carnegie Mellon Univ./Software Eng. Inst., pp. 21-32, 2004; www.sei.cmu.edu/pub/documents/04.reports/pdf/04tr018.pdf.
6. C. Woody, *Eliciting and Analyzing Quality Requirements: Management Influences on Software Quality Requirements*, tech. note CMU/SEI-2005-TN-010, Carnegie Mellon/Software Eng. Inst., 2005; www.sei.cmu.edu/publications/documents/05.reports/05tn010.html/.
7. J. Allen, "Informational Security as an Institutional Priority," 2005; www.cert.org/archive/pdf/info-sec-ip.pdf/.
8. C. Alberts, A. Dorofee, and J. Allen, *OCTAVE Catalog of Practices, Version 2.0*, tech. report CMU/SEI-2001-TR-020, Carnegie Mellon Univ., 2001; www.cert.org/archive/pdf/01tr020.pdf/.
9. C. Woody and J. Minter, "Embedding Security into a Software Development Methodology," ID: SPC0562, Abstract and Presentation for the Security Professionals Conf., Educause Resource Center, 2005; www.educause.edu/LibraryDetailPage/666?ID=SPC0562/.

Carol Woody *is a senior member of the technical staff at Carnegie Mellon University's Software Engineering Institute. Her research interests include software design and development for operational security, information and software assurance for systems of systems, and business-process survivability analysis. Woody has a PhD in information systems from Nova Southeastern University. She is a member of the IEEE, the ACM, and the Project Management Institute (PMI). Contact her at cwoody@sei.cmu.edu.*

Christopher Alberts *is a senior member of the technical staff at Carnegie Mellon University's Software Engineering Institute. His research interests include risk management, systems engineering, information security, and process improvement. Alberts has an ME in mechanical engineering from Carnegie Mellon University. He is a member of the IEEE. Contact him at cja@sei.cmu.edu.*